

### Remarks

Reconsideration of the application is requested.

Claims 1-84 have been rejected. Claims 10, 12-13 18-20, 25-28, 30, 38, 42, 52, 54-55, 60-62, 67-70, 72, 80, 82, and 84 have been amended. Claims 24 and 66 have been cancelled. Accordingly, Claims 1-23, 25-65, and 67-84 remain pending in the application.

### Claim Rejections – 35 U.S.C. § 103

To establish obviousness under 35 U.S.C. § 103, the Examiner must view the invention as a whole. Further, the Examiner is to perform the obviousness analysis in accordance with the standard set forth by the Supreme Court in *Graham v. John Deere Co.* That standard requires that the Examiner (1) determine the scope and content of the prior art; (2) ascertain the differences between the prior art and the claims in issue; (3) resolve the level of ordinary skill in the art; and (4) evaluate evidence of secondary considerations. 383 U.S. 1, 17-18 (1966); *see also* MPEP 2141. Secondary considerations include whether the invention met with commercial success, whether the invention answered a long felt need, and whether others attempting the invention have failed. *Graham*, 383 U.S. at 17-18. Further, in applying the *Graham* framework, the Examiner must consider the invention as a whole, without the benefit of hindsight. MPEP 2141.

1. In “Claim Rejections – 35 USC § 103,” item 3 on page 2 of the above-identified final Office Action, claims 1-2, 10, 30-32, 35-36, 38-39, 41-44, 52, 72-74, 77-78, 80-81, and 83-84 have been rejected as being unpatentable over U.S. Patent No. 6,253,252 to *Schofield* (hereinafter “Schofield”) under 35 U.S.C. § 103(a).

#### A. *Claims 1-2, 10, 43-44, and 52*

Claim 1 recites a “method of specifying an asynchronous web service within a procedural programming environment, the method comprising:

providing a source code representation of at least a portion of web service logic,  
the logic including at least one method declared to be a callback method;  
and  
identifying a member variable declared to implement said callback method to  
cause a compiler to generate a client proxy object for interacting  
asynchronously with the client using said callback method, and to assign  
the client proxy object to said member variable.”

Thus, when Claim 1 is viewed as a whole, as required by law, it is directed towards a novel method of specifying an asynchronous web service in a procedural programming environment. The method includes specifying at least a portion of the logic of the web service in a procedural program, and declaring a variable in the procedural program to correspond to a callback method of the asynchronous web service. The declaration of the variable is to cause a compiler incorporated with teachings of the invention to generate a proxy object for a client device to facilitate asynchronous interaction between the client and the web service, using the callback method.

In contrast, Schofield does not teach or suggest the generation of a proxy object to interact with a client in response to the specifying of a member variable declared to implement a callback method. Rather, Schofield is simply directed toward enabling asynchronous responses to distributed object calls between client and server applications communicating through a common execution environment (CEE). The client application may utilize a stub function reflecting the object interface to provide the client-side of the CEE with a service request including a reference to the object, input parameters, and a completion routine address where the client application will accept results of the service request. The CEE may then create a proxy handle structure for the service request, the proxy handle including the object reference and information regarding calls made to the same object. Upon creating the proxy handle, the service request is transferred from the client-side CEE to the server-side CEE. The server-side CEE then selects the appropriate method of the server application to fulfill the service

request. That method may fulfill the request itself or call another method to fulfill the request, synchronously or asynchronously. The results of the server application method are then received by the server-side CEE and transferred to the client-side CEE, which may use the completion routine address stored in the proxy handle associated with the request to notify the client application of the results. Both the client and server applications and the stub functions may be generated by a code generator from source files.

Thus, the only callback method arguably disclosed by Schofield as part of the web service logic (source files of the server application) is the response function. The callback identifier associated with the response function however, does not teach or suggest a “member variable declared to implement said callback method to cause a compiler to generate a client proxy object for interacting asynchronously with the client using said callback method.” The callback identifier does not cause the code generator to generate any sort of proxy object, much less a client proxy object for interacting asynchronously with the client. In fact, Schofield teaches away from such a client proxy object by its use of the CEE. It is the CEE that enables the server application to interact with the client, not a client proxy object. By providing the client and server applications with the CEE, Schofield renders the client proxy object recited by claim 1 totally unnecessary. Even if one were to assume for the sake of argument, against the plain reading of Schofield, that the CEE is a client proxy object (a point with which Applicants most strongly disagree), the CEE is not generated in response to the presence or absence of a member variable, but is provided as a separate component not even generated by the code generator of Schofield.

Also, the parameter array cited by the Examiner as suggesting a member variable is not disclosed as being declared to implement a callback function and the presence or absence of the array does not cause or prevent the generating of any sort of client proxy object. The array simply stores error information and input parameters. The code generator of Schofield does not even compile source files including the array, as the array is a creature of the CEE. Thus, it is impossible to imagine how the array

suggests a “member variable declared to implement said callback method to cause a compiler to generate a client proxy object for interacting asynchronously with the client using said callback method.”

Accordingly, claim 1 is patentable over Schofield under 35 U.S.C. §103(a).

Claims 10, 43, and 52 recite limitations similar to those of claim 1. Accordingly, for at least the same reasons, claims 10, 43, and 52 are patentable over Schofield under 35 U.S.C. §103(a).

Claims 2 and 44 depend from claims 1 and 43, incorporating their limitations, respectively. Accordingly, for at least the same reasons, claims 2 and 44 are patentable over Schofield under 35 U.S.C. §103(a).

*B. Claims 30-32, 35-36, 38-39, 41-42, 72-74, 77-78, 80-81, and 83-84*

Amended claim 30 recites, in “a server offering a web service, a method comprising:

generating by the server a request to another external web service using a proxy object previously generated by a compiler based upon a service description file associated with the external web service, wherein the request includes a callback address to identify a location to which the external web service should return a response;  
transmitting by the server the request as a request message to the external web service using one or more transmission protocols; and  
receiving by the server an asynchronous response from the external web service.

Thus, when Claim 30 is viewed as a whole, as required by law, it is directed towards a novel method of generating requests to an external web service by a proxy object of a server having a web service, the proxy object previously generated by a compiler based on a service description file associated with the external web server.

In contrast, Schofield fails to teach or suggest a proxy object previously generated based on a service description file of an external web server. Schofield, as described in part above in item 1, part A of this Response, teaches that a response function of a server application that may be associated with a callback identifier to receive asynchronous responses from an external web server. In embodiments where the server application method calls another asynchronous service to fulfill the request, the server application may include a response function to receive callback from the other asynchronous service, and may associate the response function with a callback identifier to enable the server application to continue operations while waiting for the asynchronous response.

Even if one assumes for the sake of argument that Schofield might suggest some sort of proxy object to listen for asynchronous responses, nothing suggests that such an object need be previously generated, and previously generated based on a service description file of the external web server. Schofield does not even mention such service description files or contemplate their use as a basis for a proxy object. Further, Schofield explicitly teaches that the server application generates the requests to the external asynchronous web server, thus teaching away from a previously generated proxy object making such a request.

Accordingly, claim 30 is patentable over Schofield under 35 U.S.C. §103(a).

Claims 38, 72, and 80 recite limitations similar to those of claim 30. Accordingly, for at least the same reasons, claims 38, 72, and 80 are patentable over Schofield under 35 U.S.C. §103(a).

Claims 31-32, 35-36, 39, 41-42, 73-74, 77-78, 81, and 83-84 depend from claims 30, 38, 72, and 80, incorporating their limitations, respectively. Accordingly, for at least the same reasons, claims 31-32, 35-36, 39, 41-42, 73-74, 77-78, 81, and 83-84 are patentable over Schofield under 35 U.S.C. §103(a).

2. In “Claim Rejections – 35 USC § 103,” item 26 on page 9 of the above-identified final Office Action, claims 3-8, 11, 45-50, and 53 have been rejected as being unpatentable over Schofield in view of U.S. Patent No. 6,230,160 to *Chan* (hereinafter “Chan”) under 35 U.S.C. § 103(a).

As stated above in item 1, part A of this Response, Schofield fails to teach or suggest required, recited operations of the present invention, as claimed in claims 1, 10, 43, and 52. Chan fails to cure these deficiencies. Thus, even when combined with Chan, the cited art fails to suggest novel features that are noted when the invention of claims 1, 10, 43, and 52 is viewed as a whole.

Claims 3-8, 11, 45-50, and 53 depend from claims 1, 10, 43, and 52, respectively, incorporating their limitations. Consequently, claims 3-8, 11, 45-50, and 53 are patentable over the combination of Schofield and Chan.

3. In “Claim Rejections – 35 USC § 103,” item 35 on page 11 of the above-identified final Office Action, claims 9 and 51 have been rejected as being unpatentable over Schofield in view of Chan, and further in view of U.S. Patent No. 6,209,018 to *Ben-Shachar* (hereinafter “Ben-Shachar”) under 35 U.S.C. § 103(a).

As stated above in item 2 of this Response, Schofield and Chan, alone or in combination, fail to teach or suggest required, recited operations of the present invention, as claimed in claims 1 and 43. Ben-Shachar fails to cure these deficiencies. Thus, even when combined with Ben-Shachar, the cited art fails to suggest novel features that are noted when the invention of claims 1 and 43 is viewed as a whole.

Claims 9 and 51 depend from claims 1 and 43, respectively, incorporating their limitations. Consequently, claims 9 and 51 are patentable over the combination of Schofield, Chan, and Ben-Shachar.

4. In “Claim Rejections – 35 USC § 103,” item 39 on page 12 of the above-identified final Office Action, claims 12-23, 33-34, 37, 40, 54-65, 75-76, 79, and 82 have been rejected as being unpatentable over Schofield in view of U.S. Patent Publication No. 2002/0099738 to *Grant* (hereinafter “Grant”) under 35 U.S.C. § 103(a).

*A. Claims 12-19 and 54-61*

Claims 12 and 54 recites limitations similar to those of claims 1 and 43. Accordingly, for at least the same reasons given above in item 1, part A of this Response, claims 12 and 54 are patentable over Schofield under 35 U.S.C. §103(a).

Thus, Schofield fails to teach or suggest required, recited operations of the present invention, as claimed in claims 12 and 54. Grant fails to cure these deficiencies. Thus, even when combined with Grant, the cited art fails to suggest novel features that are noted when the invention of claims 12 and 54 is viewed as a whole. Consequently, claims 12 and 54 are patentable over the combination of Schofield and Grant.

Claims 13-19 and 55-61 depend from claims 12 and 54, respectively, incorporating their limitations. Consequently, claims 13-19 and 55-61 are patentable over the combination of Schofield and Grant.

*B. Claims 20-23 and 62-65*

Amended claim 20 recites a “method for specifying web service logic within a procedural programming environment for receiving a callback from an external web service, the method comprising:

identifying a member variable to be used for interacting with said external web service;

providing a method associated with said member variable, the method having a signature and containing logic for receiving said callback from said external web service; and  
specifying one or more declarative annotations in association with said member variable to cause a compiler to generate one or more persistent components to maintain conversational state related to the external web service.”

Thus, when amended claim 20 is viewed as a whole, as required by law, it is directed towards a novel method of specifying both a callback method associated with a variable of web service logic, the callback method capable of receiving callback from an external server, and “one or more declarative annotations in association with said member variable to cause a compiler to generate one or more persistent components to maintain conversational state related to the external web service.”

At best, Schofield, described above in greater detail in item 1 of this Response, teaches a server application including a response function for receiving callback from an asynchronous web service, the response function associated with a callback identifier, wherein the logic specifying the server application and the included response function and callback identifier are compiled by a code generator. Nothing in Schofield teaches or suggests declarative annotations to cause the compiler to generate persistent components to maintain conversational state. If Schofield suggests generating any such persistent components, it suggests no more than doing so in response to logic specifying the components, as is the common practice in the art. To reach the next step, generating the components based on declarative annotations rather than simply based on logic specifying the components, one must use impermissible hindsight. The entire idea of declarative annotations, as described in the present application, is to save developers from having to specify implementation details of the web service and simply focus on specifying the service itself. Schofield, in contrast, does not disclose or reference any sort of similar “shortcut” saving developer time and reducing the complexity of code, or even recognize these problems. Accordingly, one skilled in the



art might take from Schofield the suggestion of specifying the logic of a proxy object to receive callback, but not any sort of suggestion to specify such an object using declarative annotations.

In further contrast, Grant simply discloses a method and system for integrating back-end enterprise applications with web clients. Grant achieves this, in part, through client requests to a web server, the web server having a servlet for communicating with a back-end XML engine, including receiving responses from the XML engine. Even assuming for the sake of argument that the servlet receiving responses from the XML engine suggests a proxy engine adapted to receive callback from an external web server (an assumption with which the Applicants strongly disagree that need not be addressed at this time), nothing in Grant suggests specifying declarative annotations to cause a compiler to generate persistent objects to maintain a conversational state. Even if the proxy object and persistent object are simultaneously read upon by the servlet, nothing remotely suggests specifying the servlet by any other means than that usually used in the art: specifying the full logic of the servlet. Any advantage specifying the servlet through declarative annotations might provide can only be seen through impermissible hindsight.

Accordingly, amended claim 20 is patentable over Schofield and Grant, alone or in combination, under 35 U.S.C. §103(a).

Amended claim 62 recites limitations similar to those of amended claim 20. Accordingly, for at least the same reasons, amended claim 62 is patentable over Schofield and Grant, alone or in combination, under 35 U.S.C. §103(a).

Claims 21-23 and 63-85 depend from amended claims 20 and 62, incorporating their limitations, respectively. Accordingly, for at least the same reasons, claims 21-23 and 63-85 are patentable over Schofield and Grant, alone or in combination under 35 U.S.C. §103(a).

C. *Claims 33-34, 37, 40, 75-76, 79, and 82*

As stated above in item 1, part B of this Response, Schofield fails to teach or suggest required, recited operations of the present invention, as claimed in claims 30, 38, 72, and 80. Grant fails to cure these deficiencies. Thus, even when combined with Grant, the cited art fails to suggest novel features that are noted when the invention of claims 30, 38, 72, and 80 is viewed as a whole.

Claims 33-34, 37, 40, 75-76, 79, and 82 depend from claims 30, 38, 72, and 80, respectively, incorporating their limitations. Consequently, claims 33-34, 37, 40, 75-76, 79, and 82 are patentable over the combination of Schofield and Grant.

5. In "Claim Rejections – 35 USC § 103," item 63 on page 18 of the above-identified final Office Action, claims 24-29 and 66-71 have been rejected as being unpatentable over Schofield in view of Grant, and further in view of Chan under 35 U.S.C. § 103(a).

The rejections of claims 24 and 66 are obviated by their cancellations.

As stated above in item 4, part B of this Response, Schofield and Grant, alone or in combination, fail to teach or suggest required, recited operations of the present invention, as claimed in amended claims 20 and 62. Chan fails to cure these deficiencies. Thus, even when combined with Chan, the cited art fails to suggest novel features that are noted when the invention of amended claims 20 and 62 is viewed as a whole.

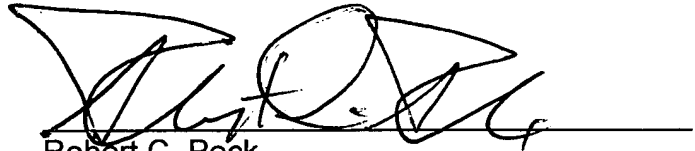
Claims 25-29 and 67-71 depend from amended claims 20 and 62, respectively, incorporating their limitations. Consequently, claims 25-29 and 67-71 are patentable over the combination of Schofield, Grant, and Chan.

Conclusion

In view of the foregoing, reconsideration and allowance of claims 1-23, 25-65, and 67-84 are solicited. Applicant submits that claims 1-23, 25-65, and 67-84 are in condition for allowance. Accordingly, a Notice of Allowance is respectfully requested. If the Examiner has any questions concerning the present paper, the Examiner is kindly requested to contact the undersigned at (206) 407-1513. If any fees are due in connection with filing this paper, the Commissioner is authorized to charge the Deposit Account of Schwabe, Williamson and Wyatt, P.C., No. 50-0393.

Respectfully submitted,

SCHWABE, WILLIAMSON & WYATT, P.C.

A handwritten signature in black ink, appearing to read 'Robert C. Peck', is written over a horizontal line.

Robert C. Peck  
Reg. No. 56,826

Dated: August 29, 2006

Schwabe, Williamson & Wyatt, P.C.  
Pacwest Center, Suites 1600-1900  
1211 SW Fifth Avenue  
Portland, Oregon 97204  
Telephone: 503-222-9981